8-2020

Anomaly Detection in Videos through Deep Unsupervised Techniques

Parikshit Prashant Shembekar
pps5251@rit.edu

Anomaly Detection in Videos through Deep Unsupervised Techniques

by

Parikshit Prashant Shembekar

A Thesis submitted to the
B. Thomas Golisano College of Computing and Information Sciences
Department of Computer Science
in partial fulfillment of the requirements for the
Master of Science
in
Computer Science
at the Rochester Institute of Technology
August 2020

Anomaly Detection in Videos through Deep Unsupervised Techniques

APPROVED BY
SUPERVISING COMMITTEE:

_____
Dr. Ifeoma Nwogu, Advisor

_____
Dr. Linwei Wang, Reader

_____
Dr. Matthew Hoffman, Observer

Abstract

Identifying abnormality in videos is an area of active research. Most of the work makes extensive use of supervised approaches, even though these methods often give superior performances the major drawback being abnormalities cannot be conformed to select classes, thus the need for unsupervised models to approach this task. We introduce Dirichlet Process Mixture Models (DPMM) along with Autoencoders to learn the normality in the data. Autoencoders have been extensively used in the literature for feature extraction and enable us to capture rich features into a small dimensional space. We use the Stick Breaking formulation of the DPMM which is a non-parametric version of the Gaussian mixture model and it can create new clusters as more and more data is observed. We exploit this property of the stick-breaking model to incorporate online learning and prediction of data in an unsupervised manner. We first introduce a two-phase model with feature extraction through autoencoders in the first step and then model inference through the DPMM in the second step. We seek to improve upon this model by introducing a model that does both the feature extraction and model inference in an end-to-end fashion by modeling the stick-breaking formulation to the Variational Autoencoder (VAE) setting.

Acknowledgments

I am extremely grateful to Professor Ifeoma Nwogu for providing me the conceptual understanding of the subject matter, guiding me in the right direction throughout this work, and also for showing patience whilst I struggled all along.

I want to thank Professor Matthew Hoffman and Professor Linwei Wang for being part of my thesis committee and for providing all the help and nudging me in the right direction with their suggestions.

I want to thank my advisor Cindy Wolfer for always being available to help me throughout my time at RIT.

And finally, I want to thank my parents Prashant Shembekar and Pournima Shembekar, without whose constant support and encouragement I wouldn't have been in a position to complete this important milestone of my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

There is a growing prevalence of surveillance cameras due to the ever-increasing security and safety risks. Examples of such risks include but are not limited to fire, accidents, falls, theft, misbehaviour, etc. The CCTV cameras need constant human monitoring which is labour-intensive. In places, where manned monitoring of CCTV footage is unavailable, the culprit identification is done after the event. The real usage of CCTV would be when a warning is received to a concerned person while the suspicious activity is happening. Thus, arises the need to use artificial intelligence algorithms to accurately and autonomously detect any questionable behaviour. With such a system, it may be possible to avert the alarming situation and avoid potential damage.

One of the challenges in this domain is that the 'questionable' behaviour depends on the context of the situation. An activity that is transgressive in certain situations (like jumping over a compound wall) might be perfectly normal in other situations (like jumping over a wall in a park). Further, an anomalous behaviour cannot be conformed to a set of suspicious actions, thus any rule-based system cannot work. It is very important to detect anomalous behaviour based on the context of the situation.

1.2 Objectives

One consensus that can be reached is that the percentage of occurrence of anomalous activity is very less compared to any normal activity, we set to exploit this very nature of an anomalous activity. The goal of this thesis is to develop a model that works in a completely unsupervised, as we have stated earlier, we cannot have predefined labels for normal and abnormal as for any given situation. The model should be able to adapt its predictions in an online fashion, as we have stated that our definition of abnormality is where the number of occurrences of an event is less and is a rare event. So, if a particular event is observed for the first time, it should predict it as abnormal and multiple occurrences of the event should encourage the model to change the events prediction to normal. Thus we set to make two fold contributions, by introducing an unsupervised model and a model adapts itself in an online manner.

Chapter 2

Background

In this chapter, we will discuss the essential concepts of the Dirichlet Process Mixture Model, Autoencoder, and also review the concepts behind the traditional Variational Autoencoder, which is used as a basis to build our model of Stick Breaking Variational Autoencoder.

2.1 Autoencoder

Autoencoders are a type of neural network architecture whose task is dimensionality reduction. They work trying to reproduce their input data. While doing this, the input data is reduced to a lower-dimensional size than the input data, and then again the size is increased to the same dimensional size as the input. This helps in capturing the high features in the input space in a compact lower-dimensional space. This is particularly useful when we are dealing with large video-based input.

It is much easier to make any inference from lower dimensional data. Thus, autoencoders form a fundamental basis for our work.

Autoencoders are comprised of two sections encoders and decoders. Encoders are responsible for the learning lower-dimensional representation, usually referred to as the bottleneck layer, from the input data. While the decoders, learn the mapping from the learned bottleneck layer to the input size, whilst trying to replicate the input space.

Figure 2.1 shows an example of an autoencoder.

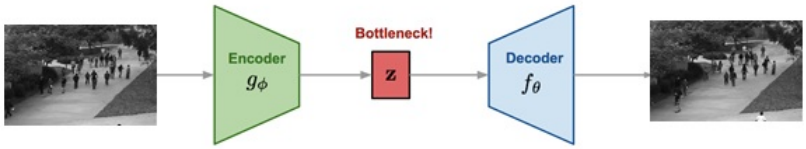

Figure 2.1: Autoencoder Architecture

2.2 Variational Autoencoder

Variational Autoencoders (VAE) [8] are class of generative models. The main idea behind VAEs is that instead of trying to reproduce a specific input, we try to generate a distribution. Instead of learning a single representation in the lower dimensional space, in VAE, two representations are learnt, signifying mean and the standard deviation of the representation. When we need a vector to pass through the decoder network, we

need to sample from the distribution.

The encoding layer is a probabilistic model given by: $q_\Phi(z|x_i)$. The decoding layer is given by another probabilistic model: $p_\Theta(x|z)$.

The loss is given by the following equation:

$L(\Theta, \phi, x, z) = \mathbb{E}_{q_\phi(z|x)}[logp_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z))$

The loss function is compromised of two parts, the reconstruction loss and the KL divergence term. The reconstruction loss is similar to an autoencoder's loss, except that there is a expectation operator because we are sampling from a distribution. The KL divergence term is used to ensure that the distribution that is generated is not too far from a normally distributed Gaussian. Since we are sampling from the distribution at the low dimensional layer, we cannot backpropagate through it. So, to be able to calculate the gradients over the entire network, we use the reparameterization trick, given by the equation: $z = \mu + \sigma \odot \epsilon$

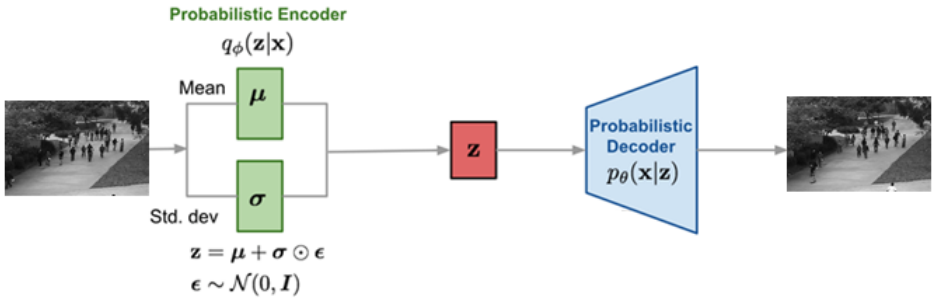

Figure 2.2: Variational Autoencoder Architecture

2.3 Gaussian Mixture Model

The Gaussian Mixture Model is a parametric model that learns a weighted sum over K Gaussian entities. Given a value of K, it creates K clusters by learning the K different means and sigmas, in a completely unsupervised manner.

2.4 Beta Distribution

It is a continous distribution between 0 and 1. It is modelled by two paramters α and β.

The probaility distribution function of Beta distribution is given by:

$$f(x) = \begin{cases} \frac{1}{B(a,b)} x^{a-1}(1-x)^{(b-1)} & if\ 0 < x < 1 \\ 0 & otherwise \end{cases}$$

Where, $B(a,b) = \int_0^1 x^{a-1}(1-x)^{b-1}dx$

2.5 Kumaraswamy Distribution

The Kumaraswampy distribution [11] is given by the following equation: $Kumaraswamy(x;a,b) = abx^{a-1}(1-x^a)^{b-1}$.

Another interesting property of the Kumaraswamy distribution is that when a=1 or b=1 or both are 1, the Kumaraswamy and Beta distributions are equivalent.

2.6 Dirichlet Process Mixture Model

The Dirichlet Process Mixture Model is a non-parametric mixture model that is, it is not bounded by the fixed K number of Gaussians. It can infer the number of distributions in the data in an unsupervised manner.

The Dirichlet Process: The Dirichlet process is a probabilistic distribution model over discrete distributions. The Dirichlet process (DP) can be specified by some base probabilistic distribution H_0 and a scaling parameter denoted as α. The α can take a value between 0 and ∞. Values closer to 0 lead to more discrete distributions while those closer to ∞ lead to a more continuous distribution.

Formally, a DP is defined for a sample H from a Dirichlet Process with parameters $H_0 : \Omega \to \mathbb{R}$ where α is a distribution over Ω. For a disjoint partition of Ω: $\Omega_1, ...\Omega_k$, and given a sample H such that $H \sim DP(H_0, \alpha)$, we have:

$(H(\Omega_1)), ..., H(\Omega_k)) \sim Dir(\alpha H(\Omega_1)), ..., \alpha H(\Omega_k))$

Ω represents our sample space, and we are taking a discrete partition over our sample space and creating a discrete distribution over our base distribution.

The Stick Breaking Method: [27] The generative process of DPMMs proceeds as follows: First we draw, $\beta_i \sim Beta(1, \alpha)$ for $i \in \mathbb{N}$. Then draw $\Theta_i \sim H_0$.

Then we construct the mixture weights π by taking $\pi_i(\beta_{1:\infty}) = \beta_i \prod_{j<i}(1-\beta_j)$. The π_i is an indicator function which evaluates to 0 everywhere ex-

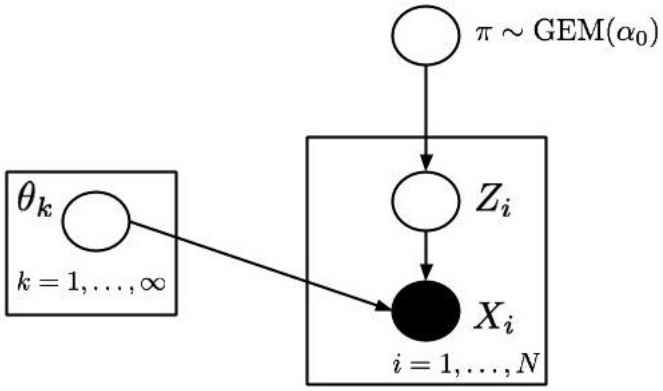

Figure 2.3: The Dirichlet Process Mixture Model

cept where $\pi_i = 1$

Then for each observation $n \in \{1, ...N\}$, we draw $z_n \sim \pi(\beta_{1:\infty})$, and then draw $x_n \sim f(\Theta_{z_n})$

2.7 Stick Breaking Representation

The stick breaking model is one of the representations of the Dirichlet Mixture Model.

The π are random independent weights drawn from a base distribution H_0 which satisfy the following two conditions: $0 \leq \pi_k \leq 1$ and $\sum \pi_k = 1$.

$$\pi_k = \begin{cases} v_1 \; if \; k = 1 \\ v_k \prod_{j<k} (1 - v_j) \; for \; k > 1 \end{cases}$$

Where v is drawn from a $Beta$ distribution such that $v_k \sim Beta(1, \alpha)$ which gives rise to the stick-breaking formulation.

Chapter 3

Related Work

Anomaly Detection in videos is a field of active research. A lot of work has been done for the past several years. Though most of the work has been based upon classical machine learning techniques using handcrafted features and most of them have been based upon supervised learning techniques. Though supervised techniques help us to achieve good results they can work only as well as the labels we have. It is not possible to have a global set of all possible cases that can be anomalous. Thus arises the need to have an unsupervised approach to detecting an anomaly.

3.1 Supervised Learning Approaches

Lavee et al. [12] create histograms of the video events for both normal and abnormal events, once these are created, the new test sample's histograms are compared to the histograms whose classes are known, to make the

predictions.

Xu et al. [29] use autoencoders (AE) to obtain the low dimensional latent representation of videos. Once the embedded layers are obtained in a supervised learning approach one-vs-all Support Vector Machines are used to classify between normal and abnormal videos. The autoencoder architecture used is a stacked autoencoder where one AE is trained on the video input as is, another AE is trained on the optical flow of the video and a third AE is trained to learn a joint representation from both the input types.

Sultani et al [24] use a weakly-supervised approach where the labels were at video-level instead of frame-level. For each video, they knew if it contained anomaly or not, the training for performed using Multiple Instance Learning (MIL) [1]. The training was performed by dividing the videos into chunks, for each individual chunk it is not known if it contains anomaly. So they use the bag-of-videos approach, where chunks of samples belonging to normal videos are grouped together likewise chunks of samples belonging to abnormal videos are grouped together. Then a pre-trained three-dimensional convolutional network is used to extract features from the video chunks, and finally, another convolutional network is applied to rank these instances into normal/abnormal samples.

A major limitation with supervised approaches in the context of abnormality detection is that in a general scenario it is not known prior, the entire set of abnormal cases that can occur, which tends to limit the

practical use-cases.

3.2 Semi - Supervised Learning Approaches

In this approach, generally, the model is trained on just the normal class and during testing, both the normal and abnormal samples are shown to the model. This kind of setting enables us to avoid defining the labels for the abnormalities and thus overcome the major drawback with supervised approaches. Although such settings do not allow for online learning of the model and limit model adaptability.

Jacob [21] uses autoencoders to obtain the embedded low-dimensional video representation, but along with the AE training, K-Means clustering is also performed in order to cluster the embedded space into clusters of known "normal" spaces. They argued that clustering the embedded space allowed better learning of the Gaussian mixture model and helped in improving the prediction score. The K-Means centers are used as initial clusters of the Expectation-Maximization algorithm which would formulate k Gaussian distributions, each representing known 'normal' space. Also to note is that the training of the AE-KMeans clustering is performed only on 'normal' instances and only in testing both the normal and abnormal instances are provided to the model.

Ravanbaksh et al [19] use a similar approach wherein they use optical flow and temporal frames, they quantize them into 7 bits and then map these bits into histograms, histograms of normal frames would conform

to a similar distribution, whereas an abnormal one would have a very different representation.

[30] compute gradients and optical flow of the normal videos. Then they compute histograms of optical flow (HOF) and histogram of gradients (HOG), using these features they build the Gaussian model. They found that increasing the number of Gaussians up to a certain value, increased the model's performance, although increasing beyond a certain point, it leads to overfitting.

Levine et al [4] use Variational Autoencoders (VAE) to learn the normal distribution of the embedded space. The KL-divergence used in VAE enables the data of similar nature to group together, thus in a way, it clusters the embedded space. To learn the temporal and spatial features they incorporate architecture similar to [29] with one AE learning the spatial features and another AE learning the temporal features through the dynamic flow. The two autoencoders give different predictions finally the scores are combined to give a joint prediction.

Another approach tackling the problem of abnormality detection is through future frame prediction, this approach has been used by [14], [16], and [25].

Tang et al [25] have used Generative Adversarial Networks [5] to predict the future frame. They argue that the generally accepted way to observe anomalies by finding the large difference in reconstruction loss is not robust. The reconstruction loss is more pronounced when a frame

is predicted, generally, a normal frame predicted would have a low reconstruction loss whereas an anomalous frame will have a reconstruction loss.

[16] also used the approach of predicting future frames. The frames are predicted using an architecture involving ConvLSTM layers [23].

Another work using ConvLSTMs is [2] in which they follow a similar approach of semi-supervised training by training on normal samples, although they do not predict the future frame, they predict abnormality by directly checking the reconstruction loss.

3.3 Unsupervised Learning Approaches

In this section, we review the unsupervised approaches in the literature. Though a good quality of work has been done, most of the work uses handcrafted features for model building and little work has been done using neural networks, which have proven to outperform classical machine learning approaches in various other applications [10].

[31] is a classical unsupervised approach to abnormality prediction. They circumvented the model building approach to solve the task by building a co-occurrence matrix. The matrix mapped the videos to prototypes, which were obtained from hand crafted features. Then abnormalities were found by finding similarities between the matrix space.

Wang et al [28] introduce an unsupervised approach. They use Latent Dirichlet Allocation (LDA) and Hierarchical Dirichlet Process (HDP) to

model activities. They use optical flow to get the input vectors, the optical flow captures both the magnitude and direction of motion. They compute the likelihoods of the samples from the LDA model to make the predictions.

Similarly, [18] proposed a completely unsupervised approach of both feature generation and modeling. They use Hidden Markov Models (HMM) to model the trajectories of motion in videos. Another unsupervised technique [6] which again uses HMMs to model the motion trajectories. They incorporate clustering of the trajectories and then these trajectories are fit using an HMM, they repetitively perform this step, that is after an HMM is fit, the data is again clustered and a new HMM is fit on the data. This helped them to avoid overfitting the data.

Roshtkhari [20] used spatiotemporal video volumes (SVT). These SVTs are then clustered and so that similar SVTs get grouped together. They model these SVTs using a probabilistic distribution while each data point being a single SVT or a group, arguing that a group of SVTs allow for capturing context across the long temporal dimension. To predict abnormalities they develop a codebook of the SVTs with each codeword being an SVT. A prior probability is assigned to a codeword based on its earlier occurrences.

Recently, [3] showed that Deep Generative Models often fail to detect abnormalities in data. They showed that the abnormalities can be captured by using Bayesian deep generative models. In this approach they

rather than sampling from the decoder parameters, they infer the posterior distribution by using probabilistic measures over the parameters of the generative network.

Chapter 4

Datasets

4.1 Datasets

We will be performing our tests on these datasets: UCSD Ped1, UCSD Ped2, Avenue Dataset, Shanghai Dataset and our own Custom Dataset

In the UCSD datasets, all the training videos provided are of normal instances and each test video consists of both normal and anomaly instances.

The UCSD datasets are of people walking across a walkway, and any instance of a cyclist, skater, or a car passing through the walkway is an anomaly.

The Ped1 dataset has 6800 frames of training data all of the normal instances. While it has 2000 frames of pixel-level labeled test frames. The Ped2 dataset has 2550 training frames and 2010 test frames with

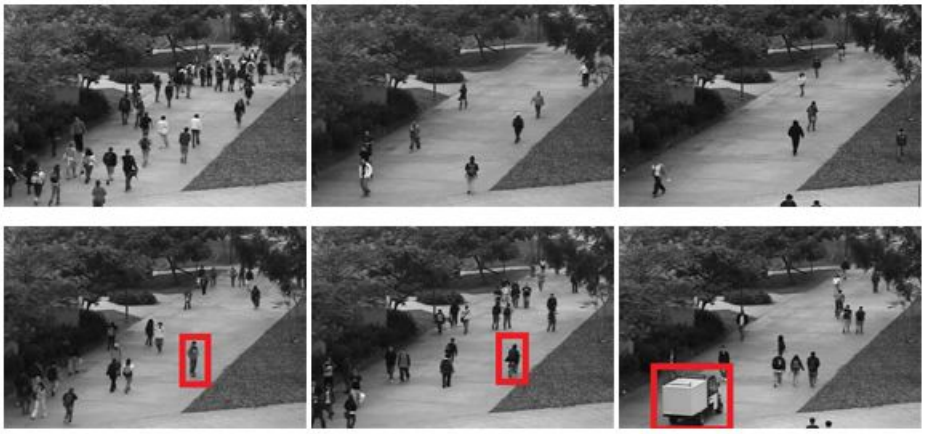

Figure 4.1: PED1 dataset samples

pixel-level labels.

Figure 4.2: PED2 dataset samples

Figure 4.1 and Figure 4.2 shows the examples of UCSD Ped1 and UCSD Ped2 respectively. The upper rows show the examples of normal samples and the bottom rows show instances of abnormalities, the

Figure 4.3: Avenue dataset samples

abnormal regions are denoted with bounding red box.

The Avenue dataset is a more challenging one. It has 15328 training frames and 15324 testing frames. The abnormality examples include a person running, dropping and throwing a bag in the air, and people moving in the direction.

Figure 4.4: Shanghai dataset samples [14]

The Shanghai dataset would be by far the most challenging dataset to work on, as it has 130 different types of abnormalities. It has 274,515

training frames and 42,883 test frames. Also, pixel-level labels are available for all test frames.

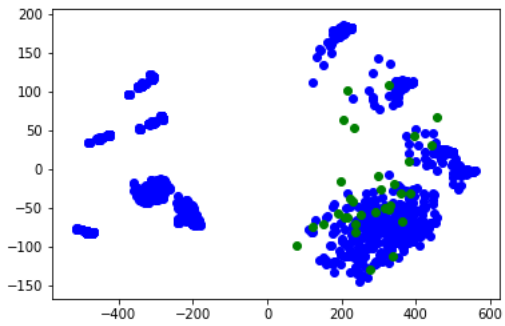

Figure 4.5: PCA on PED2 dataset, Showing the challenge of abnormality detection in videos.

Figure 4.5 shows the PCA plot obtained of the PED 2 dataset, the plot motivates the need for a model that can classify the abnormalities from the normal samples. It depicts the overwhelming imbalance in the classes of normal (blue samples) and abnormal (green samples). The plot also demonstrates why a classifier like SVM may not be the best approach to detect abnormalities in video samples as even in the principal components with most variance, the abnormalities seem to be overlaid on the normal samples.

Chapter 5

Methodology

5.1 Input Format

We train on cuboids of data which are sub-samples of the videos, specifically our input data is processed to be of shape 32x32x20. Where the height and width are 32 and the depth obtained by concatenating consecutive frames is 20. Which chose our depth of the cuboid to be 20, as on an average video input, it enables us to capture about one second worth of input. We decided to choose 32x32 size because it is a standard size used in popular computer vision tasks, example CIFAR dataset [9] has images of size 32x32 and LeNet [13] also expects the same size of input.

Figure 5.1 visually shows how cuboids are formed. The cuboidal samples help in capturing both the spatial and temporal information

and alleviates the computational complexity incurred upon using optical flow. We are able to use spatio temporal cuboids by making use of 3d convolutions [26].

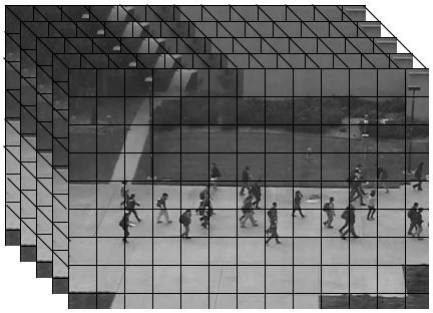

Figure 5.1: Spatio-Temporal Samples

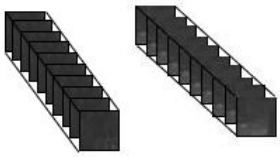

Figure 5.2: Sample of input to Autoencoder

5.2 Two stage Dirichlet Process Model

The proposed approach expands on the work of [21] wherein we make use of the fact that activities captured in videos can be conformed to a specific group of activities, thereby, clustering the data allows us to find patterns of normal behavior. Thus the general framework is as follows: We perform unsupervised feature extraction using autoencoders and these

extracted features are passed on to a non-parametric Dirichlet Process Mixture Model, specifically the Stick Breaking implementation [22]. The weights of a stick breaking model adapt to the input data and the number of sticks (weights) can theoretically grow to infinite as a new type of data is observed.

In this approach, we divide our training process into two steps. In the first step, we use an autoencoder simply acts as a feature extractor, and in the next step, we train a Dirichlet process mixture model on the extracted features.

Figure 5.3 shows the autoencoder architecture used. The autoencoder acts as a feature extractor and learns the low dimensional representation of the input data. The Stick breaking model is then trained on these extracted features.

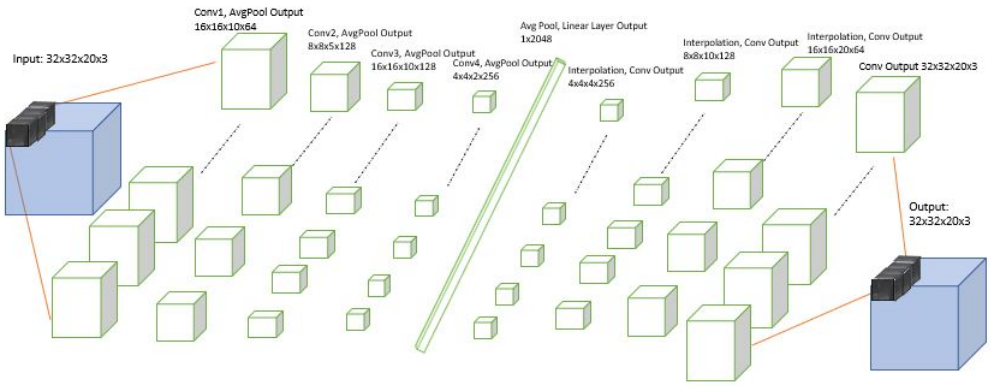

Figure 5.3: Feature extraction using autoencoder

5.3 Dirichlet Process Model - Variational Autoencoder

We attempted to improve upon the two-stage model by proposing a single-stage end-to-end model, which performs feature extraction and fit the Dirichlet process mixture model as the neural net is getting trained.

The theoretical definition of the model is based on the work by [17]. Figure 5.4 shows the architecture of our proposed model.

In this, unlike the traditional VAE, samples are not drawn from a Gaussian Distribution, rather they are drawn from a GEM distributuon. The latent samples are sampled from the generative model using the following equation: $\pi_i \sim GEM(\alpha_0)$. Where π is the vector of stick breaking weights. α_0 is the concentration parameter of the GEM distribution.

5.4 Implementation Details

All our neural nets were run on 8 Tesla T4 GPUs parallelly. We used Adam optimizer [7] for our gradient descent optimization. We used the Mean Squared Error loss function for both our models. The quantitative metrics we use are Area Under the Curve (AUC). Regarding the threshold value, in the plots, we are not reporting an accuracy based on a single threshold value, rather we are using multiple thresholds, to create the (receiver operating characteristic (ROC) curve and then computing AUC

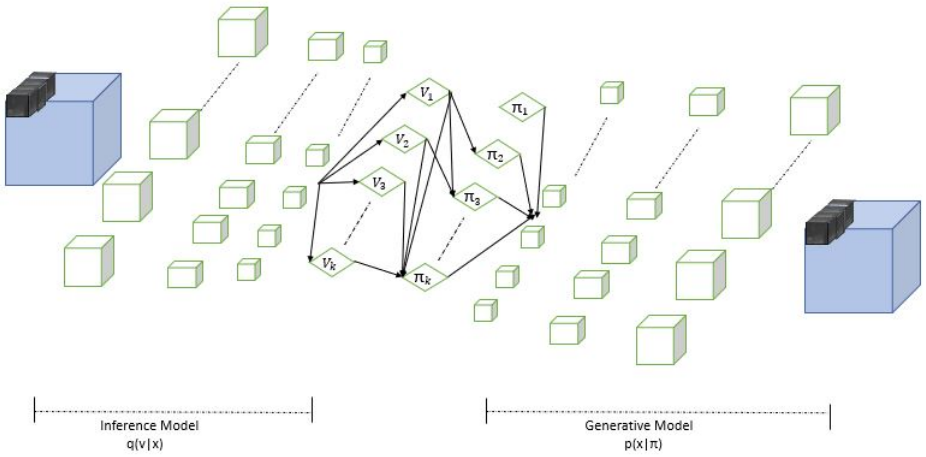

Figure 5.4: Proposed Architecture of Stick Breaking Variational Autoencoder

which lies between 0 and 1. This is the only quantitative metric we report, as others have done in the literature.

Our procedure to plot the ROC: We find the likelihoods for all the samples normal and abnormal. We then calculate the average likelihood for all the normal samples and the average likelihood for all the abnormal samples. The likelihood scores for the abnormal samples are always less than that of the likelihood of the normal samples.So, we take a very small jump of 0.05 and vary our threshold from the average abnormal score to average normal likelihood score, and in each iteration increase the threshold by this jump value. During each iteration, we examine the likelihood of the sample; if it is greater than the threshold we predict it as normal and if it is less than the threshold, we predict it as abnormal.

Then, for that iteration, we calculate the count of true positives, false positives, true negatives, and false negatives. Based on these values, we calculate the true positive rate (TPR) and the false positive rate (FPR). Each TPR and FPR represents a point for our ROC curve. Once the entire ROC curve is generated, the AUC score is calculated using all the TPR and FPR values.

Chapter 6

Results

We have obtained pixel-level accuracies for the two-stage model and the Stick Breaking Variational Autoencoder (SB-VAE) on the UCSD Pedestrian datasets, Avenue dataset, and the Shanghai dataset.

The pixel-level evaluation strategy is followed as mentioned in [15]. This works in the following manner, if more than 40% of the pixels detected as anomalous overlap with the ground truth, the patch is termed as true positive.

For the two-stage model, we first train the autoencoder on normal videos to learn the low dimensional feature representation. Once we get the low dimensional representation we pass these to the Dirichlet Mixture Model which maps the data to a non-parametric Gaussian representation of the features. After this entire process, our model training is completed. We then pass the test video samples to the autoencoder to

extract the features and these extracted features are then passed to the trained Dirichlet mixture model to get the log-likelihood scores. Since the Dirichlet model is trained on normal samples, the log-likelihood for the normal test samples tends to be higher and the log-likelihood for the abnormal samples is lower. We then use some threshold value in between these values, to predict whether a sample is normal or abnormal.

For the SB-VAE model, the low dimensional feature learning and the mapping to learn to the non-parametric Dirichlet Mixture Model is a simultaneous process. The Kumaraswamy distribution learns the μ, from the low dimensional features. Once, the model is well trained, that is the loss is low, the autoencoder has learned a good low dimensional representation of the data. We then use this learned mu to get an average μ from all the training samples. Again, we are training only on normal videos. During testing, we get μ, for each test sample, and we measure the distance of the test sample's μ to the learned average training μ. For normal samples, this distance is less and for the abnormal samples, this distance is more. We then use some threshold value to predict each sample to abnormal or normal.

6.1 Results on PED1 dataset

The plot shown in figure 6.1 demonstrates the likelihood values of the normal and abnormal samples. It shows that most of the normal samples have high likelihood values whereas the likelihood significantly drops for

real abnormal samples.

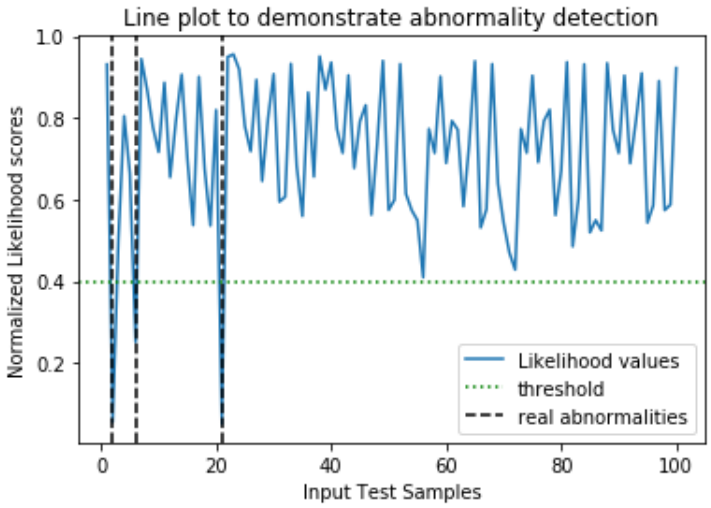

Figure 6.1: Likelihood Plot on PED1

Table 6.1: AUC scores comparison on PED1

	Pixel-level AUC scores
AnomalyNet	0.452
Detection at 150 FPS	0.638
AMDN	0.67
GMM Two stage	0.74
GMM VAE	0.71
Our Two Stage Model	**0.77**
Our SBVAE	0.66

6.2 Results on PED2 dataset

We have tested our two-stage model on PED2 dataset and we are comparing our model against that of [21] whose pixel wise on PED2 dataset is the highest with 0.79 AUC. Figure 6.2 show our ROC plot against the ROC of [21].

Our Model has AUC of 0.82 on PED2 dataset. At least for PED2 dataset our model out performs [4] (0.78 AUC) and [21].

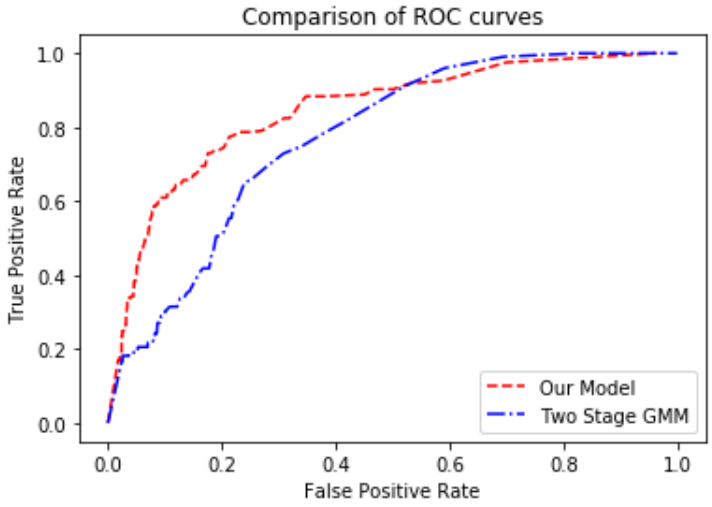

Figure 6.2: ROC Comparisson on PED2

6.3 Results on Avenue dataset

Table 6.3 shows the pixel-level performance of our models on the Avenue dataset.

Table 6.2: AUC scores comparison on PED2

	AUC scores
AnomalyNet	0.528
GMM Two stage	0.79
GMM VAE	0.78
Our Two Stage Model	**0.82**
Our SBVAE	**0.79**

Table 6.3: AUC scores comparison on Avenue

	AUC scores
AnomalyNet	0.94
Our Two Stage Model	0.72
SBVAE	0.63

6.4 Results on Shanghai dataset

Shanghai dataset is relatively a very new dataset. In the literature only frame level accuracies are available on the Shanghai dataset.

Table 6.4: Pixel-level AUC scores comparison on Shanghai

	AUC scores
Our Two Stage Model	0.62
Our SBVAE	0.55

Table 6.5: Frame level AUC scores on Shanghai Dataset

	AUC scores
Future Frame Prediction	0.728
Our Two Stage Model	0.67
Our SBVAE	0.58

6.5 Online learning results

We have trained our two-stage model on the PED 1 dataset and we are testing on the PED 2 dataset. As each sample from the PED 2 is evaluated from the Dirichlet Process Mixture Model, we also fit the new sample to the previously trained model on the PED 1 dataset. Figure 6.3 demonstrates how the likelihood plot drops initially, as the model is earlier trained on PED1 and prediction begins on PED2. Although, as we go on training on PED2 the likelihood values for PED2 normal data increase to near 1. The likelihood values do drop when an abnormal sample is seen.

We have demonstrated our results qualitatively on the two-stage model. We show that for normal samples the likelihood samples are near 1 and for abnormal samples, the likelihood values drop and are relatively closer to 0.

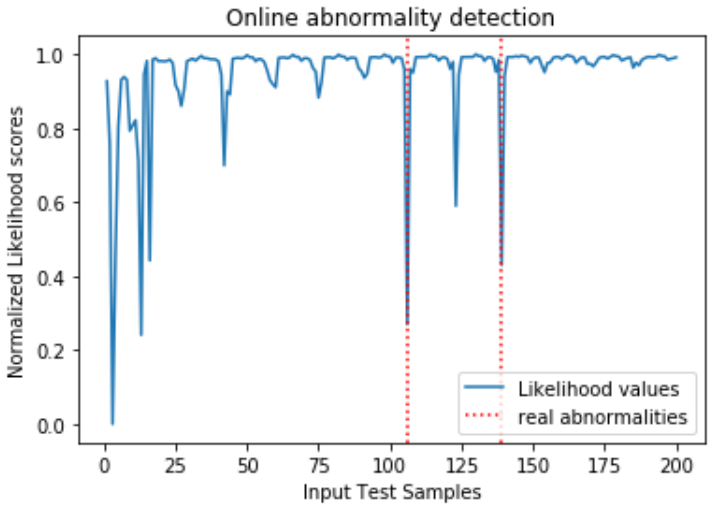

Figure 6.3: Online Learning of PED2

We wanted to check the performance on PED1 after fitting it to the PED2 dataset. We see that the likelihood scores of PED1 tend to vary a lot more than that of the PED2. Though the difference between the normal likelihood scores and abnormal likelihood scores is still large. This has been shown in figure 6.4

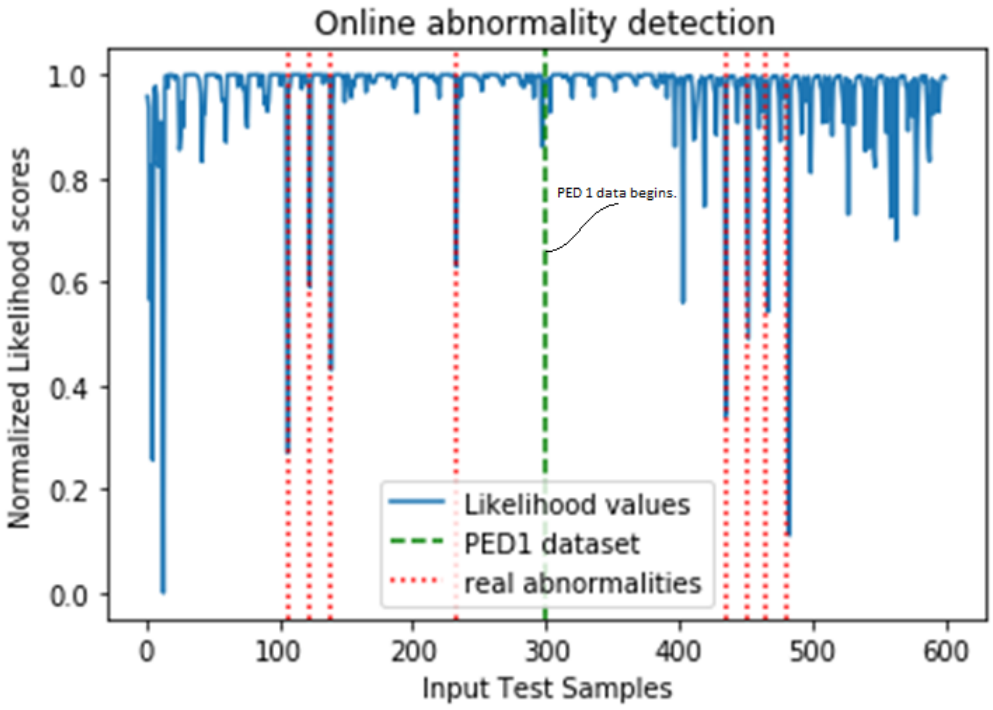

Figure 6.4: Re-checking performance on PED1

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we have proposed two approaches for the anomaly identification in videos. The Dirichlet Process Mixture Model (DPMM) two-stage model built upon the ideas from the literature review, shows to perform as well as the state of the art models in the literature for the pixel-level evaluation. We have validated this claim by performing our evaluations on several datasets. We also attempted at adapting the two-stage process into an end-to-end model. Although we have not been able to reproduce the performance of the two-stage DPMM model on our Stick Breaking VAE model.

7.2 Future Work

We feel that the results of the Stick Breaking VAE can be further improved upon. One way would be to perform an ablation study on the size of the input spatio-temporal cuboid, hyper-parameter tuning of the stick-breaking parameters could further help in performance enhancement. We have worked with the MSE loss, autoencoders tend to slightly perform better when used with the SSIM loss function, this could be another criterion to experiment with.

As suggested in the paper [3] it would be interesting to try stochastic gradient MCMC optimization over the maximum likelihood.

For the online learning methodology that we demonstrated, while fitting the PED2 samples we also have to fit the PED1 samples that we have trained on earlier. We have to do this because no standard API implementing Dirichlet Process Mixture (DPMM) provides a 'partial-fit' function that allows us to build on the model we new data. A good future work would be to implement this function for the DPMM.

Bibliography

[1] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 577–584. MIT Press, 2003.

[2] Yong Shean Chong and Yong Haur Tay. Abnormal event detection in videos using spatiotemporal autoencoder, 2017.

[3] Erik Daxberger and José Miguel Hernández-Lobato. Bayesian variational autoencoders for unsupervised out-of-distribution detection, 2019.

[4] Yaxiang Fan, Gongjian Wen, Deren Li, ShaoHua Qiu, and Martin D. Levine. Video anomaly detection and localization via gaussian mixture fully convolutional variational autoencoder. *CoRR*, abs/1805.11223, 2018.

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua

Bengio. Generative adversarial networks, 2014.

[6] F. Jiang, Y. Wu, and A. K. Katsaggelos. A dynamic hierarchical clustering method for trajectory-based unusual video event detection. *IEEE Transactions on Image Processing*, 18(4):907–913, 2009.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[8] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[9] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, NYU, 2009.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[11] P. Kumaraswamy. A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1):79 – 88, 1980.

[12] Gal Lavee, Latifur Khan, and Bhavani Thuraisingham. A framework for a video analysis tool for suspicious event detection. *Multimedia Tools Appl.*, 35:109–123, 08 2007.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition, 1998.

[14] W. Liu, D. Lian W. Luo, and S. Gao. Future frame prediction for anomaly detection – a new baseline. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[15] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1975–1981, 2010.

[16] Jefferson Ryan Medel and Andreas Savakis. Anomaly detection in video using predictive convolutional long short-term memory networks, 2016.

[17] Eric T. Nalisnick and Padhraic Smyth. Stick-breaking variational autoencoders. *arXiv: Machine Learning*, 2017.

[18] Kan Ouivirach, Shashi Gharti, and Matthew N. Dailey. Incremental behavior modeling and suspicious activity detection. *Pattern Recognit.*, 46:671–680, 2013.

[19] Mahdyar Ravanbakhsh, Moin Nabi, Hossein Mousavi, Enver Sangineto, and Nicu Sebe. Plug-and-play cnn for crowd motion analysis: An application in abnormal event detection, 2016.

[20] M. J. Roshtkhari and M. D. Levine. Online dominant and anomalous behavior detection in videos. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2611–2618, 2013.

[21] Martin D. Levine Seby Jacob. Anomaly detection from videos : A deep learning approach. *MS Thesis McGill University*, 2018.

[22] Jayaram Sethuraman. A constructive definition of the dirichlet prior. *Statistica Sinica*, 4:639–650, 01 1994.

[23] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai kin Wong, and Wang chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting, 2015.

[24] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos, 2018.

[25] Yao Tang, Lin Zhao, Shanshan Zhang, Chen Gong, Guangyu Li, and Jian Yang. Integrating prediction and reconstruction for anomaly detection. *Pattern Recognition Letters*, 129:123 – 130, 2020.

[26] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks, 2014.

[27] Pyro.ai Uber. Dirichlet process mixture models in pyro. `https://pyro.ai/examples/dirichlet_process_mixture.html#The-Stick-Breaking-Method-(Sethuraman,-1994)`, 2017.

[28] X. Wang, X. Ma, and W. E. L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):539–555, 2009.

[29] Dan Xu, Elisa Ricci, Yan Yan, Jingkuan Song, and Nicu Sebe. Learning deep representations of appearance and motion for anomalous event detection. *CoRR*, abs/1510.01553, 2015.

[30] Y. Yuan, Y. Feng, and X. Lu. Statistical hypothesis detector for abnormal event detection in crowded scenes. *IEEE Transactions on Cybernetics*, 47(11):3597–3608, 2017.

[31] Hua Zhong, Jianbo Shi, and Mirkó Visontai. Detecting unusual activity in video. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'04, page 819–826, USA, 2004. IEEE Computer Society.